Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

iRecSpot-EF: Effective sequence based features for recombination hotspot prediction

Md Rafsan Jani, Md Toha Khan Mozlish, Sajid Ahmed, Dewan Md Farid, Swakkhar Shatabda*

Department of Computer Science and Engineering, United International University, Madani Avenue, Satarkul, Badda, Dhaka, 1212, Bangladesh

ARTICLE INFO

Keywords:
Recombination hotspots
Novel feature extraction
DNA sequences
Classification
Feature selection
Web application

ABSTRACT

In genetic evolution, meiotic recombination plays an important role. Recombination introduces genetic variations and is a vital source of biodiversity and appears as a driving force in evolutionary development. Local regions of chromosomes where recombination events tend to be concentrated are known as hotspots and regions with relatively low frequencies of recombination are called coldspots. Predicting hotspots and coldspots can enlighten structure of recombination and genome evolution. In this paper, we proposed a predictor, called iRecSpot-EF to predict recombination hot and cold spots. iRecSpot-EF uses a novel set of features extracted from the genome sequences. We introduce the frequency of (l, k, p)-mers in the sequence as features. Our proposed feature extraction method hinges solely upon the nucleotide sequences, thus being cost-effective and robust. After feature extraction, the most informative features are selected using AdaBoost algorithm. We have selected logistic regression as the classification algorithm. iRecSpot-EF was tested on a standard benchmark dataset using cross-fold validation. It achieved an accuracy of 95.14% and area under Receiver Operating Characteristic curve (auROC) of 0.985. The performance of iRecSpot-EF is significantly better than the state-of-the-art methods. iRecSpot-EF is readily available for use from http://iRecSpot.pythonanywhere.com/server. All relevant codes are available via open repository at: https://github.com/mrzResearchArena/iRecSpot.

1. Introduction

0Recombination is the process where two DNA molecules exchange nucleotide sequences with each other. In the context of meiosis, two homologous chromosomes exchanges large portions of their DNA during recombination. Hotspots are genomic regions with relatively higher frequencies of recombination and coldspots are with relatively lower frequencies of recombination [1]. Double-strand DNA is required to break for crossovers to take place [2]. Recombination hotspot plays a vital part in evolutionary development. The study of recombination hotspot gives useful insights into the basic function of inheritance and the study of genetic diversity. Recombination provides knowledge about DNA sequence variation and patterns along human chromosomes and this may help to map the position of alleles that cause various diseases [3,4]. They are also helpful in revealing locations of single nucleotide polymorphisms.

Recombination hotspots are often identified by modelling linkage disequilibrium [5]. However, recent trend in the literature of computational methods has formulated the problem of predicting recombination hotspots in a genome sequence as a supervised learning problem. Many successful classification algorithms like Support Vector Machine (SVM) [6], Random Forest (RF) [7], Ensemble Classifiers [8], etc have been employed for the task. Most of the methods in the literature are dependent on sequence information or nucleotide contents. However, simple nucleotide contents [9] do not take into account the sequence order information, and thus the prediction ability is limited. Among other successful features are gapped k-mers [7], pseudo-nucleotide composition [10], physical and thermodynamic properties of DNA sequences [11], etc. It is to be noted that patterns that exist in DNA sequences are enormous in number and its quite challenging to find effective patterns or features for hotspot prediction.

RF-DYMHC [7], iRSpot-GAEnsC [8] and IDQD [9] are proposed predictors all built using features from sequence content information. RF-DYMHC [7] used gapped dinucleotide composition features. IDQD [9] combined increment of diversity with quadratic discriminant analysis based on k-mer frequencies in DNA sequences. However, both of these methods avoided sequence order information. iRSpot-PseDNC [10], iRSpot-DACC [12], iRSpot-DACC-PCA [12], iRSpot-EL [13] and iRSpot-TNCPseAAC [14] were proposed addressing this problem. However, there was still scope for further improvements which came in

* Corresponding author.
E-mail addresses: rafsanjani.muhammod@gmail.com (M.R. Jani), mmozlish141089@bscse.uiu.ac.bd (M.T. Khan Mozlish), sahmed133002@bscse.uiu.ac.bd (S. Ahmed), dewanfarid@cse.uiu.ac.bd (D.M. Farid), swakkhar@cse.uiu.ac.bd (S. Shatabda).

https://doi.org/10.1016/j.compbiomed.2018.10.005
Received 8 June 2018; Received in revised form 7 October 2018; Accepted 7 October 2018
0010-4825/

the form of the following methods. iRSpot-PseDNC [10] proposed a SVM model based using pseudo dinucleotide composition. iRSpot-DACC [12] is also a SVM model which adopted Principal Component Analysis (PCA). iRSpot-TNCPseAAC [14] was based on DNA trinucleotide composition and the corresponding pseudo amino acid components [15]. iRSpot-EL [13] further improved the performances by combining pseudo K-tuple nucleotide composition (PseKNC) and dinucleotide based auto-cross covariance (DACC) into an ensemble learning approach. Pse-in-One [16] is a stand-alone tool which facilitates the generation of 14 modes of features for DNA sequences which can be broadly divided into three categories such as Nucleic acid composition (Basic kmer), Autocorrelation(Dinucleotide-based auto-covariance,Trinucleotide-based autocovariance) and Pseudo nucleotide composition(PseKNC, PseDNC). Pse-in-One 2.0 [17] is an improvement over Pse-in-One that incorporated 6 new feature-modes for DNA sequences such as Moran autocorrelation (MAC), Geary autocorrelation (GAC), increment of diversity (IDKmer) [18] etc. repDNA [19] developed a python package with almost similar functionalities as Pse-in-One above generating 15 features for DNA sequences. A recent predictor iRSpot-ADPM [20] was proposed that mainly focuses on extracting different features between di-nucleotide pairs at different positions in DNA sequence.

In this paper, we propose iRecSpot-EF, a novel prediction method which stands for **i**dentification of **Rec**ombination hot**Spot**s using **E**ffective **F**eatures. Our method uses a novel feature extraction method. We use (l, k, p)-mer frequencies in a given sequence as a features along with others. We apply an AdaBoost based feature selection technique to reduce unnecessary features and selected logistic regression as our classification algorithm. iRecSpot-EF is tested on standard benchmark dataset using cross-fold validation. Experimental results show that iRecSpot-EF significantly outperforms existing state-of-the-art methods and achieves highest accuracy. We have also implemented a webserver based on our proposed method and its ready to be used from http://iRecSpot.pythonanywhere.com/server.

2. Materials and methods

The system diagram of our proposed method is given in Fig. 1. In the training phase, iRecSpot-EF starts with the DNA sequences in the given dataset. Using the feature extraction method, it creates the original dataset which then undergoes subsequent feature selection. After feature selection only a few number of features are selected and the original dataset is transformed into a reduced one. A classifier model is learned on the reduced dataset and the model is saved for future prediction and testing purposes. In the testing phase, for a given DNA sequences the features are generated and passed through the model that generates a prediction. The model is also used for the implementation of the webserver. Rest of this section describes the various components of iRecSpot-EF.

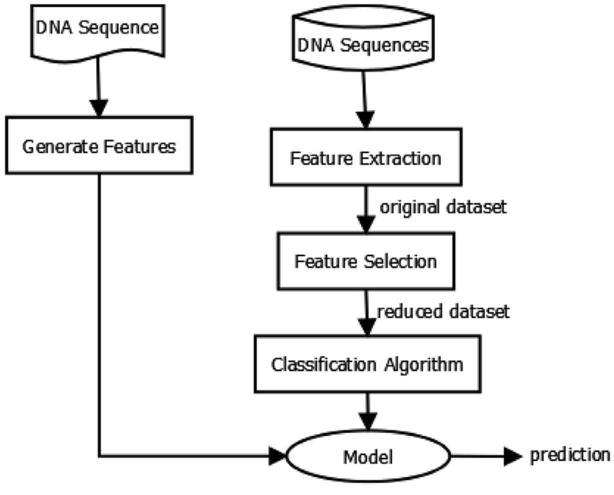

Fig. 1. System Diagram for iRecSpot-EF.

2.1. Benchmark dataset

In this paper, we have used the widely accepted and benchmark dataset for recombination hotspot prediction in the literature [7,12–14]. For a binary classification problem such as the recombination hotspot prediction, the dataset is formulated as follows:

$$\mathbb{S} = \mathbb{S}^{+} \cup \mathbb{S}^{-} \tag{1}$$

Here, dataset $\mathbb{S}$ is an union of set of positive instances $\mathbb{S}^{+}$, those with hotspots and the set of negative instances $\mathbb{S}^{-}$ those with coldspots. The initial dataset contained 490 positive instances or hotspots and 591 negative instances. The positive instances were selected for which the hybridization ratio was higher than 1.5 [7] and negative instances were selected as such when the hybridization ratio was less than 0.82. CD-HIT [21] was used to remove homologous sequences with similarity greater than 75% to reduce redundancy. The reduced dataset contained 478 positive instances and 572 negative instances. The dataset is quite balanced.

2.2. Feature extraction

0Each sequence in the dataset is a composition of nucleotides. In this paper, we have generated sequence-based features using the different components of those nucleotides. Our target was to capture the sequence content and order information as much as possible and also to identify significant patterns in the sequences that can discriminate hotspots from the coldspots. Five set of features were generated. This section gives an overview of how they were generated.

2.2.1. (l, k, p)-mer composition

A number of composition based features are proposed in the literature of protein attribute and DNA sequence functionality prediction. Our motivation in this paper was to find effective nucleotide patterns that will help to make decision retaining both content and order information. We call this feature (l, k, p)-mer composition. To demonstrate (l, k, p)-mer composition, we need to define (l, k, p)-mer first. (l, k, p)-mers are generalization or fusion of k-gapped di-nucleotide compositions. Formally, (l, k, p)-mer is a string of length l with k consecutive gaps starting at position p in the string. For example, AC_T is a (l, k, p)-mer, with $l = 4, k = 1, p = 3$. If we take all possible (l, k, p)-mers with $l = 4, k = 1, p = 3$ there will be 64 such(l, k, p)-mers. In this way, simple tri-nucleotides are (l, k, p)-mers with $l = 3, k = 0$ where p is not important since $k = 0$. Note that, simple k-gapped (l, k, p)-mers with $p = 1$, where l and k will vary. Thus (l, k, p)-mers provide generalization over simple l-mers and gapped l-mers. (l, k, p)-mers are the patterns that we look for into the given DNA sequences. (l, k, p)-mer composition is defined as follows:

$$(l, k, p) - \text{mer Composition} = \frac{1}{L}\text{Count}((l, k, p) - \text{mer}) \tag{2}$$

Here, L is the length of the DNA sequence and we have generated (l, k, p)-mer compositions from the sequences for $1 \leq l \leq 20, 0 \leq k \leq 15$ and $0 \leq p < l - k$. Further explanation on (l, k, p)-mer generation is provided as Supplementary Information Sp2.

2.2.2. Cumulative skew

Due to deamination process there is a difference of the count of G and T in forward and reverse strands. The forward strand often have more G and T. The cumulative skew [22] is defined formally as:

$$GC\ skew = \frac{\sum G - \sum C}{\sum G + \sum C};\ AT\ skew = \frac{\sum A - \sum T}{\sum A + \sum T} \tag{3}$$

Here as $\sum A$ represents the total number of A, $\sum C$ represents the total number of C from the sequence and so forth. Two features were generated using the cumulative skew method.

2.2.3. Z-curve

Z-curve theory [23] is often used in genomic sequence analysis [24,25]. It has got three components in three axis. They are defined as following.

$$\begin{cases} x\ axis = (\sum A + \sum G) - (\sum C + \sum T) \\ y\ axis = (\sum A + \sum C) - (\sum G + \sum T) \\ z\ axis = (\sum A + \sum T) - (\sum G + \sum C) \end{cases} \tag{4}$$

Three features were generated using the Z-Curve method.

2.2.4. GC content

In general, GC-content is expressed as a percentage value (%) [26].

$$GC\ Content = \frac{\sum G + \sum C}{\sum A + \sum C + \sum G + \sum T} \times 100\% \tag{5}$$

DNA with high GC-content is more stable than DNA with low GC-content. One feature was generated using the GC-content method.

2.2.5. AT/GC ratio

Single feature was generated using the AT/GT Ratio method. The equation is given below.

$$AT/GCRatio = \frac{\sum A + \sum T}{\sum G + \sum C} \tag{6}$$

A summary of the features used in this paper is given in Table 1.

2.3. Feature selection

Though we have extracted a large number of features from the sequences, these features are all sequence based and thus very easy to generate and not time consuming compared to structure based or physico-chemical properties based features and pseudo-nucleotide compositions. However, reduction in the number of features often results in methods with a better accuracy of prediction due to the diminished likelihood of model over-fitting [12,27,28]. Several methods are used in the literature of classification problem to handle the feature reduction issue: principal component analysis [12], recursive feature elimination [28], etc. In this paper, we have used a wrapper method employing AdaBoost algorithm for selecting potent features. Firstly, we have run AdaBoost ensemble classifier with decision trees [29] on the dataset. Then each estimator of the ensemble classifier provides with significance or importance of the features that it was created with. We then selected only those features that are significant. Note that, gapped compositions with different number of nucleotides in each side offers a pattern (l,k,p)-mer composition which is the most effective of all features. We suppose this is the most effective of all the features and clearly distinguish our method from others such as in Ref. [20]. Also note that this particular type of feature selection by AdaBoost is computationally very cheap compared to other methods.

Table 1
Summary of features used by iRecSpot-EF.

Feature Types	No. of Features
$(l,\ k,\ p)$-mer composition	40,672
Cumulative skew	2
Z-Curve axis	3
GC content	1
AT/GC Ratio	1
Total	40,679

2.4. Logistic regression

Logistic regression is one of the most elegant and popular classification algorithms [30]. Logistic regression is a linear classifier that tries to separate the positive instances from the negative instances using a straight line. Logistic regression due to its simplicity is popular and does not overfit the data in general. For a given number of features, $x_1,\ x_2,\ \cdots, x_n$, the binary label or prediction y is a function of the given features. The prediction in logistic regression is simply defined by the equation below:

$$y = \sigma(w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n) \tag{7}$$

Here $w_0,\ w_1,\ \cdots, w_n$ are the coefficients or weights associated with different features and $\sigma(z)$ is the logistic regression decision function. The most popular of the function is the sigmoid function as defined below:

$$\sigma(z) = \frac{1}{1 + exp(-z)} \tag{8}$$

Logistic regression is usually trained with gradient descent optimization algorithm to find the model parameters or coefficients. Moreover, we have utilized L2 regularization for keeping model overfitting in check.

2.5. Performance evaluation

A large number of evaluation metrics are used in the literature of classification methods to evaluate performances of different methods [31]. In this paper, we have adopted the mostly used measures and metrics used in the literature of recombination hotspot prediction: accuracy (Acc), sensitivity (S_n), specificity (S_p), Mathew's Correlation Coefficient (MCC), F1-Score, area under precision-recall curve (auPR) and area under receiver operating characteristic curve (auROC).

For any binary classification problem accuracy (Acc) is the percentage of number of correctly classified instances to the total number of instances. Another important metric is the sensitivity (S_n) or recall which is the true positive rate or number of correctly classified positive instances to the total number of positive examples. Similar is specificity (S_p) or true negative rate. Precision is similar to these and is defined by the ratio of number of true positives to the total number of positive predictions. F1-Score is the harmonic mean of precision and recall. For all these measures the range of values are in [0,1]. A higher value of these metrics indicate a better performing predictor. Mathew's correlation coefficient (MCC) is another measure of the quality of the classification model. It varies from -1 to $+1$ which respectively denotes negative classification correlation and positive classification correlation. Probabilistic classifiers such as logistic regression often depends on thresholds to decide the classification labels. For such cases, area under the precision recall curve (auPR) and area under the receiver operating characteristic curve (auROC) are important measures. They reveal the strength of the underlying classifier regardless of the threshold chosen.

Along with these metrics, it is very important to choose sampling techniques for classification algorithms [32]. Most common are independent test sets and cross-validations. In this paper, we have used cross-validation since they are robust, reduces over-fitting and widely used in the literature of recombination hotspot prediction and thus suitable for comparison of different algorithms.

3. Results and discussion

In this section, we present all the experimental results achieved in this study and relevant analysis. All the experiments were done in a Computing Machine provided by CITS (Center of IT Services), United

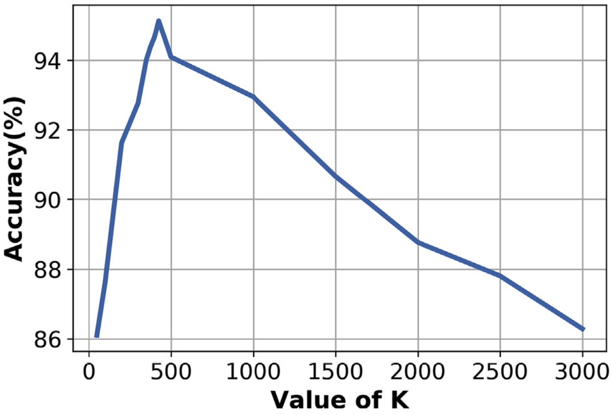

Fig. 2. Plot of accuracy against different values of K during feature selection.

International University. The machine was equipped with 8 core processors each core having a Dell R 730 Intel Xeon Processor (E5-2630 V3) with 2.4 GHz speed and 18.5 GB memory. The programs were written in Python language using Python 3.5 and Scikit-learn library of Python. All experiments were run 10 times and the average results are reported only.

3.1. Feature ranking and selection

Feature selection in this paper is done using ranking by an AdaBoost classifier. A critical decision in the feature selection procedure is to decide the parameter K. Here, after the ranking of the features, we have to determine the number of features we want to retain in the feature selection stage. We have done so using a 70: 30 train-test split initially. An AdaBoost model was trained on 70% of the instances for feature ranking and remaining 30% instances were utilized for scrutinizing the potency of k highest ranked features. In our experiments, we varied the value of K in a wide range.

The plot of accuracy achieved by selecting different values of K is given in Fig. 2. From these experiments, we have decided that the best value of K for our dataset is $K = 425$. Moreover, significance value for all the deselected features is 0.000 for our dataset which further reinforces our choice of the number of features to select,K. Observation above can be utilized for using AdaBoost classifier for feature selection in a mechanical manner without having to decide upon the number of features manually for the dataset in hand. A list of the features along with their significance values are provided as Supplementary Information Sp1. Table 2 gives a summary of feature significance in sorted order of ranking for the first 425 features.

Note that, we have employed 10-fold cross validation for reporting the results of the classifiers. We have performed feature selection procedure mentioned above at the beginning of each cross-validation iteration using the train-set only, trained a classification model using the selected features and employed unseen test-set for evaluation of the trained model. We have done so for avoiding any biased estimation of model performance.

Table 2
Summary of feature significance and ranking in feature selection.

Ranks of Feature	Significance per feature
1–3	0.008
4–10	0.006
11–62	0.004
63–425	0.002

3.2. Classification algorithm selection

After feature selection, we performed rigorous experiments on the dataset for selecting the most robust classification algorithm for our feature set. Ten different classifiers were investigated in our experiments: Decision Tree (DT) [33], K-Nearest Neighbor (KNN) [34], Naive Bayesian Classifier (NB) [35], Linear Discriminant Analysis (LDA) [36], Random Forest (RF) [37], Bagging [33], AdaBoost [38], Gradient Boosting (GBoost) [39], Support Vector Machines (SVM) [40] and Logistic Regression (LR) [30]. We trained these classifiers with 10-fold cross-validation on the dataset to measure their effectiveness in identifying hotspots and coldspots, using the feature sets extracted and then selected using the methods above. The summary of the results in terms of sensitivity (S_n), Specificity (S_p), F1-Score, MCC, Accuracy (Acc), auPR and auROC is given in Table 3.

The best values achieved in this experiment are shown in bold faced fonts in Table 3. In terms of accuracy, worst performing classifier was Naive Bayesian Classifier. K-Nearest Neighbor algorithm with $k = 7$ was among the poor performers in terms of sensitivity. However, KNN classifier was among the best performers in terms of Specificity. It gives us an idea of the distribution of the instances in the feature space. Decision tree classifier was run using Gini Impurity as the criterion for attribute selection and was in the lower end in terms of performance. SVM classifier was run using Radial Basis Function kernel. We used L2 regularization for avoiding over-fitting in Logistic Regression. SVM and Logistic Regression (LR) classifiers outperformed all other classifiers in terms of all the metrics. In terms of specificity and auPR SVM is slightly better performing than the LR classifier. However, we selected LR as the best performing classifier and suitable for our method as its accuracy was highest of 95.14% and superior to SVM in terms of MCC and F1-Score which are 0.9037 and 0.9465 respectively.

We have also drawn a box plot of accuracy of different classifiers as shown in Fig. 3. From the box plot, it is also evident that considering the error bars, LR and SVM are superior in performance compared to the rest of the classifiers explored in this paper. Ensemble classifiers like Gradient Boosting, AdaBoost, Random Forest and Baggin Classifiers are in between the best of worst performing classification algorithms. We have also shown the analysis of area under ROC curve (auROC) for all these classifiers. The plot is shown in Fig. 4. The results achieved by the classifiers can be accredited mainly to the nature of the reduced feature space that aptly preserves the underlying trends in the unmodified feature space while reducing the risk of over-fitting which results in increased accuracy over the test-set.

3.3. Comparison with other methods

From the previous experiments, we selected top 425 features and Logistic Regression classifier for our method iRecSpot-EF. In this section, we compare the performance of iRecSpot-EF with that of the already proposed state-of-the-art predictors in the literature. We have chosen eight previous predictors for the purpose of comparison. They are: RF-DYMHC [7], IDQD [9], RSpot-PseDNC [10], iRSpot-TNCPseAAC [14], iRSpot-DACC [12], iRSpot-EL [13], iRSpot-ADPM1575 [20] and iRSpot-ADPM [20]. Note that, all these methods done their experiments on the same benchmark dataset that we are using in this paper. Results in terms of Sensitivity (S_n), Specificity (S_p), Accuracy (Acc) and MCC are reported in Table 4. Note that, we have not re-run their methods to report the results in Table 4. These are the results taken as reported in their papers. From the results shown in Table 4, it is evident that our proposed method iRecSpot-EF significantly outperforms all these methods in terms of all the evaluation metrics considered for the experiments and, classification methods and feature selection approaches employed by other state-of-the-art predictors is also given in 4.

Table 3
Comparison of performances of different classification algorithms on the benchmark dataset.

Classifiers	S_n(%)	S_p(%)	F1-Score	MCC	Acc(%)	auPR	auROC
DT	70.92%	75.00%	0.7049	0.4601	73.15%	0.6323	0.7297
KNN (k = 7)	67.15%	94.93%	0.7735	0.6583	82.28%	0.9000	0.9210
NB	85.77%	49.83%	0.6978	0.3765	66.18%	0.5968	0.7060
LDA	82.64%	83.57%	0.8170	0.6621	83.14%	0.8946	0.9134
RF	67.15%	88.46%	0.7398	0.5752	78.76%	0.8404	0.8722
Bagging	71.76%	90.03%	0.7787	0.6344	81.72%	0.8685	0.8823
AdaBoost	81.59%	86.89%	0.8259	0.6890	84.47%	0.9206	0.9161
GBoost	79.08%	91.08%	0.8335	0.7127	85.62%	0.9307	0.9256
SVM	90.38%	**98.43%**	0.9397	0.8968	94.76%	**0.9889**	0.9846
LR	**94.35%**	95.80%	**0.9465**	**0.9037**	**95.14%**	0.9877	**0.9850**

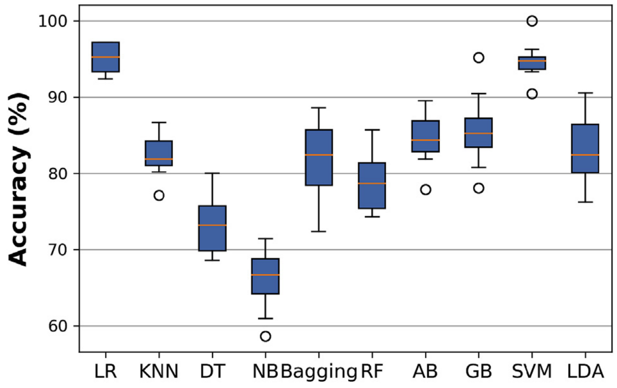

Fig. 3. Box plot showing the accuracy of different classification algorithms.

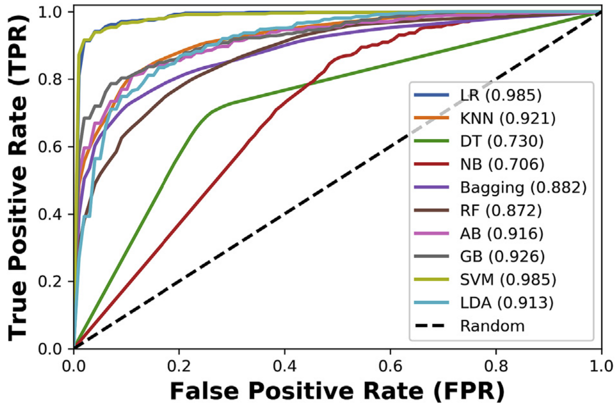

Fig. 4. Receiver Operation Characteristic cuve for different classifiers achieved during cross-fold validation.

3.4. Webserver implementation

We have implemented an easy to use server for iRecSpot-EF and made it available from http://iRecSpot.pythonanywhere.com/server. The web interface takes a sequence as input in FASTA format and provides predictions on whether it is a hotspot or coldspot. However, we would like to assert that confident predictions are desirable. Hence, a threshold value has been hand-pick(0.70 for our system), and a sequence is predicted as hotspot or coldspot only if the classifier can back its prediction with a probability greater than the threshold. Here, the probability value serves as a measure of prediction-confidence and any prediction unable to satisfy the desired confidence-threshold has been tagged as **None**. Moreover, we have provided the users of the webserver with the option of choosing a window-size for inspecting whether particular portions of the input sequence are hotspots or coldspots. Further details on the webserver are provided as Supplementary Information Sp3. For the benefit of the community and researchers, we have also made our codes available via open repository at: https://github.com/mrzResearchArena/iRecSpot.

3.5. Whole genome analysis

We have also experimented on yeast chromosome III. Chromosome-III is 316,620 bp long. We have used three different window sizes, $W = 500$, $W = 1000$ and $W = 1500$ and we have found predictions for respective points as probability by our predictor iRecSpot-EF. Fig. 5 shows plot of the probabilities along with the normalized recombination count determined experimentally by Mancera et al. [41]. The effectiveness of iRecSpot-EF can be noted from the similarity of the predictions with that of the laboratory methods.

4. Conclusion

In this paper, we have proposed a predictor, iRecSpot-EF to efficiently identify recombination spots employing a set of novel features in combination with a coherent feature selection approach. In total, forty thousand six hundred and seventy-nine (40,679) features were generated by the newly proposed feature generation method. Our features

Table 4
0Comparison of performance, classification methods and feature selection approaches employed by other state-of-the-art predictors.

Methods	S_n(%)	S_p(%)	MCC	Acc(%)	Classification Method	Feature Selection Approach
RF-DYMHC	73.01%	86.56%	0.6049	80.40%	RF	None
IDQD	79.52%	81.82%	0.6160	80.77%	SVM	None
iRSpot-PseDNC	71.75%	85.84%	0.5830	79.33%	SVM (RBF kernel)	None
iRSpot-TNCPseAAC	76.56%	70.99%	0.4737	73.52%	SVM (RBF kernel)	None
iRSpot-DACC	75.71%	88.16%	0.6470	82.52%	SVM	PCA
iRSpot-EL	75.29%	88.81%	0.6510	82.65%	Ensemble Approach	Clustering Approach
iRSpot-ADPM1575	74.88%	90.04%	0.6613	83.14%	SVM	Weights calculated by SVM
iRSpot-ADPM	77.19%	90.73%	0.6905	84.57%	SVM	Weights calculated by SVM
iRecSpot-EF	**94.35%**	**95.80%**	**0.9037**	**95.14%**	**Logistic Regression**	**AdaBoost**

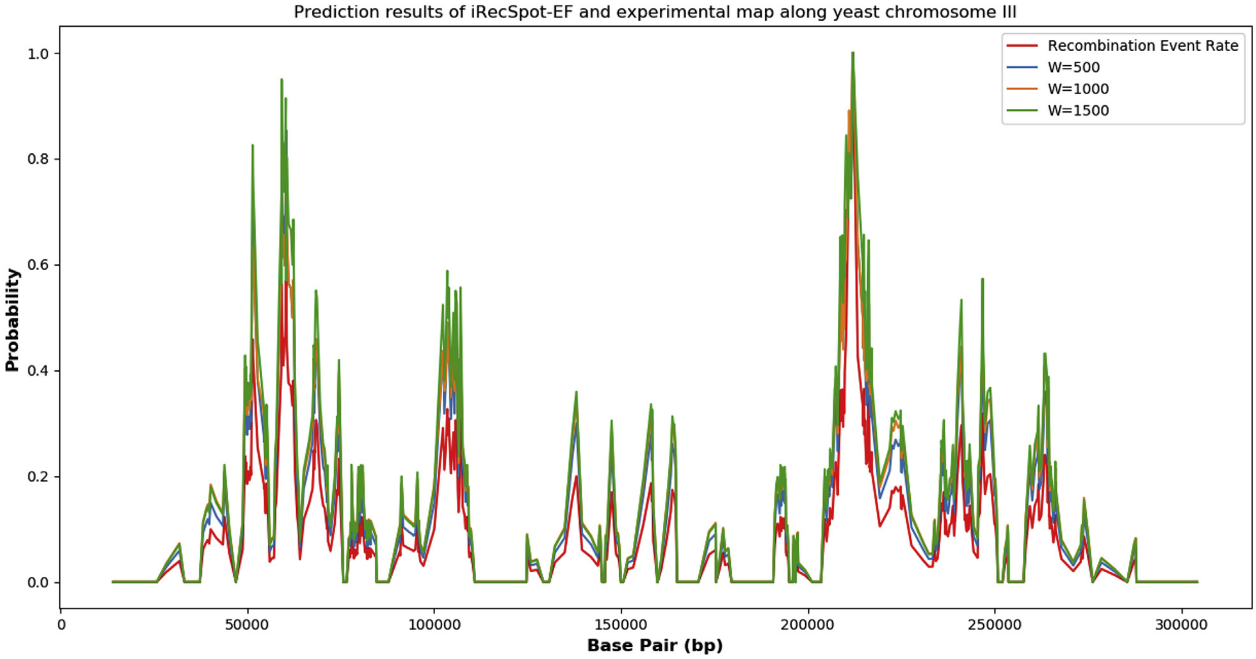

Fig. 5. Comparison between prediction results of iRecSpot-EF and experimental map along yeast chromosome III. The red line represents the recombination event rate determined experimentally by Ref. [41]. The other curves represent the probability values calculated by iRecSpot-EF with different window sizes.

include (***l,k,p***)-mers frequencies and another six(6) features were generated using Z-Curve, Cumulative skew, GC content and AT/GC ratio. Since all our features are purely sequence-based, their extraction process remains frugal while the features being an apt representation of the underlying phenomenon of interest. Most informative features were selected using AdaBoost algorithm and finally, Logistic regression was used to identify recombination spots. Our proposed (***l,k,p***)-mers extraction method has shown great potential to generate effective features from DNA sequences which has been corroborated by empirical analysis. As iRecSpot-EF accomplished to identify recombination spots better than any other existing method, it may play a significant role in genetics and cell biology. Furthermore, the simplicity and the efficiency of the method is very promising as a bioinformatics approach to reveal the mechanism of recombination and genome variation. We are trying more optimization approaches to generate features. In addition to these, we wish to add a graphical visualization similar to Fig. 5 as an extension to our web application.

Conflict of interests

The authors declare that there are no conflict of interests.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2018.10.005.

References

[1] A.J. Jeffreys, L. Kauppi, R. Neumann, Intensely punctate meiotic recombination in the class ii region of the major histocompatibility complex, Nat. Genet. 29 (2) (2001) 217.

[2] F. Baudat, J. Buard, C. Grey, A. Fledel-Alon, C. Ober, M. Przeworski, G. Coop, B. De Massy, Prdm9 is a major determinant of meiotic recombination hotspots in humans and mice, Science 327 (5967) (2010) 836–840.

[3] S.S. Abeysinghe, N. Chuzhanova, M. Krawczak, E.V. Ball, D.N. Cooper, Translocation and gross deletion breakpoints in human inherited disease and cancer i: nucleotide composition and recombination-associated motifs, Hum. Mutat. 22 (3) (2003) 229–244.

[4] J. Hey, What's so hot about recombination hotspots? PLoS Biol. 2 (6) (2004) e190.

[5] N. Li, M. Stephens, Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data, Genetics 165 (4) (2003) 2213–2233.

[6] T. Zhou, J. Weng, X. Sun, Z. Lu, Support vector machine for classification of meiotic recombination hotspots and coldspots in saccharomyces cerevisiae based on codon composition, BMC Bioinf. 7 (1) (2006) 223.

[7] P. Jiang, H. Wu, J. Wei, F. Sang, X. Sun, Z. Lu, Rf-dymhc: detecting the yeast meiotic recombination hotspots and coldspots by random forest model using gapped dinucleotide composition features, Nucleic Acids Res. 35 (suppl_2) (2007) W47–W51.

[8] M. Kabir, M. Hayat, irspot-gaensc: identifying recombination spots via ensemble classifier and extending the concept of chous pseaac to formulate dna samples, Mol. Genet. Genom. 291 (1) (2016) 285–296.

[9] G. Liu, J. Liu, X. Cui, L. Cai, Sequence-dependent prediction of recombination hotspots in saccharomyces cerevisiae, J. Theor. Biol. 293 (2012) 49–54.

[10] W. Chen, P.-M. Feng, H. Lin, K.-C. Chou, irspot-psednc: identify recombination spots with pseudo dinucleotide composition, Nucleic Acids Res. 41 (6) (2013) e68–e68.

[11] G. Liu, Y. Xing, L. Cai, Using weighted features to predict recombination hotspots in saccharomyces cerevisiae, J. Theor. Biol. 382 (2015) 15–22.

[12] B. Liu, Y. Liu, X. Jin, X. Wang, B. Liu, irspot-dacc: a computational predictor for recombination hot/cold spots identification based on dinucleotide-based auto-cross covariance, Sci. Rep. 6 (2016) 33483.

[13] B. Liu, S. Wang, R. Long, K.-C. Chou, irspot-el: identify recombination spots with an ensemble learning approach, Bioinformatics 33 (1) (2016) 35–41.

[14] W.-R. Qiu, X. Xiao, K.-C. Chou, irspot-tncpseaac: identify recombination spots with trinucleotide composition and pseudo amino acid components, Int. J. Mol. Sci. 15 (2) (2014) 1746–1766.

[15] K.-C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins: Structure, Function, and Bioinformatics 43 (3) (2001) 246–255.

[16] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, K.-C. Chou, Pse-in-one: a web server for generating various modes of pseudo components of dna, rna, and protein sequences, Nucleic Acids Res. 43 (W1) (2015) W65–W71.

[17] B. Liu, H. Wu, K.-C. Chou, Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of dna, rna, and protein sequences, Nat. Sci. 9 (04) (2017) 67.

[18] W. Chen, L. Luo, L. Zhang, The organization of nucleosomes around splice sites, Nucleic Acids Res. 38 (9) (2010) 2788–2798.

[19] B. Liu, F. Liu, L. Fang, X. Wang, K.-C. Chou, repdna: a python package to generate various modes of feature vectors for dna sequences by incorporating user-defined physicochemical properties and sequence-order effects, Bioinformatics 31 (8) (2014) 1307–1309.

[20] L. Zhang, L. Kong, irspot-adpm: identify recombination spots by incorporating the associated dinucleotide product model into chous pseudo components., J. Theor. Biol. 441..

[21] W. Li, A. Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, Bioinformatics 22 (13) (2006) 1658–1659.

[22] A. Grigoriev, Analyzing genomes with cumulative skew diagrams, Nucleic Acids Res. 26 (10) (1998) 2286–2290.

[23] R. Zhang, C.T. Zhang, Z. curves, An intutive tool for visualizing and analyzing the dna sequences, J. Biomol. Struct. Dyn. 11 (4) (1994) 767–782.

[24] R. Zhang, C.-T. Zhang, A brief review: the z-curve theory and its application in

genome analysis, Curr. Genom. 15 (2) (2014) 78–94.

[25] C.-T. Zhang, R. Zhang, Analysis of distribution of bases in the coding sequences by a digrammatic technique, Nucleic Acids Res. 19 (22) (1991) 6313–6317.

[26] M.T. Madigan, J.M. Martinko, J. Parker, et al., Brock Biology of Microorganisms vol. 13, Pearson, 2017.

[27] M. R. Uddin, A. Sharma, D. M. Farid, M. M. Rahman, A. Dehzangi, S. Shatabda, Evostruct-sub: an accurate gram-positive protein subcellular localization predictor using evolutionary and structural features, J. Theor. Biol., 443, 138–146.

[28] S.Y. Chowdhury, S. Shatabda, A. Dehzangi, Idnaprot-es: identification of dna-binding proteins using evolutionary and structural features, Sci. Rep. 7 (1) (2017) 14938.

[29] T. Hastie, S. Rosset, J. Zhu, H. Zou, Multi-class adaboost, Stat. Interface 2 (3) (2009) 349–360.

[30] D.R. Cox, The regression analysis of binary sequences, J. Roy. Stat. Soc. B (1958) 215–242.

[31] D. M. Powers, Evaluation: from Precision, Recall and F-measure to Roc, Informedness, Markedness and Correlation.

[32] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, Ijcai, vol. 14, Canada, Montreal, 1995, pp. 1137–1145.

[33] J.R. Quinlan, et al., Bagging, boosting, and c4. 5, AAAI/IAAI, vol. 1, 1996, pp. 725–730.

[34] D.T. Larose, K-Nearest Neighbor Algorithm, Discovering Knowledge in Data: an Introduction to Data Mining, (2005), pp. 90–106.

[35] I. Rish, An empirical study of the naive bayes classifier, IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, IBM, 2001, pp. 41–46.

[36] A.J. Izenman, Linear discriminant analysis, Modern Multivariate Statistical Techniques, Springer, 2013, pp. 237–280.

[37] A. Liaw, M. Wiener, et al., Classification and regression by randomforest, R. News 2 (3) (2002) 18–22.

[38] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, J. Jpn. Soc. Artif. Intell. 14 (771–780) (1999) 1612.

[39] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[40] V. Vapnik, I. Guyon, T. Hastie, Support vector machines, Mach. Learn. 20 (3) (1995) 273–297.

[41] E. Mancera, R. Bourgon, A. Brozzi, W. Huber, L.M. Steinmetz, High-resolution mapping of meiotic crossovers and non-crossovers in yeast, Nature 454 (7203) (2008) 479.